REMARKS

This Amendment is filed in response to the Office Action mailed on Oct. 5, 2006. All rejections and objections are respectfully traversed.

Claims 1-7, 9-12 and 21-51 are in the case.

Claims 1, 9, 22, 31 and 39 have been amended.

New claims 42-51 have been added.

Rejection of Claims 36-41 on Same Grounds as 1-7 and 21

At page 6 of the Office Action, rather than detail specific rejections to claims 9-12 and 22-41, the Office Action states "they comprise the same subject matter as claims 1-7 and 21 rejected above, and are therefore rejected on the same merits."

The Applicant respectfully urges that this statement is inaccurate. In particular, claims 36-41 recite identifying borders within a regular expression "***using a deterministic finite state machine,***" which is not recited in claims 1-7 or 21.

The Applicant defines a "deterministic finite state machine" as a Deterministic Finite Automation. Specifically, page 15, lines 9-12 of the Specification states "A DFA [Deterministic Finite Automation], which may ***also be referred to as a deterministic finite state machine***, is a finite state machine with exactly one transition for each given symbol and state" (emphasis added).

The combination of Fritchman and Sherman does not teach use of deterministic finite state machine (Deterministic Finite Automation) in identifying borders of regular expressions. While Fritchman mentions Deterministic Finite Automatons in its background section (*see* Fritchman col. 1, lines 50-54, 59-60), Fritchman disparages their use with the system disclosed in Fritchman, teaching away from employing them. Specifically, Fritchman states at col. 2, lines 3-13:

> The restrictive nature of the pattern language used with SQL's LIKE predicate makes it possible to derive a simpler, more efficient algorithm that does not require construction of automata and tracking of state changes for each character processed during evaluation of the predicate. While suitable for evaluation of fully general patterns based on unrestricted regular expressions, ***automata-based algorithms are overkill for the LIKE predicate. The present invention achieves significantly accelerated performance for evaluation of patterns such as those permitted by the LIKE predicate, which utilize only wild card operators*** (single-character and multi-character).

Sherman does not even mention deterministic finite state machines (Deterministic Finite Automations).

Accordingly, the two references taken together would not suggest the Applicant's novel use of the structures, in combination with all the other claim elements. The Applicant respectfully directs the Examiner's attention to **MPEP 2141.02 (VI),** which cautions against finding obviousness when references teach away from the claimed invention. **MPEP 2141.02 (VI)** cautions that "[a] prior art reference must be considered in its entirety, i.e., as a whole, including portions that would lead away from the claimed invention." The statements in Fritchman that DFA are "overkill" and that "significantly accelerated performance" may be obtained by not using them, would certainly lead one away from their use.

Further, the Applicant has added new claims 42-51 that claim in part details relating to novel uses and adaptations of DFAs. Even if the Examiner should disagree that claims 36 and 39 are allowable, the Applicant respectfully urges that dependent claims 42-51 are clearly allowable over the cited prior art. For example claim 42 reads:

> 42. The method of claim 36 wherein the deterministic finite state machine is constructed by combining separate deterministic finite state machines of each of the two or more regular expressions into a single deterministic finite state machine that identifies borders in the two or more regular expressions.

Neither reference suggest such novel features.

Claim Rejections – 35 U.S.C. §112

At pages 2-3 of the Office Action, claims 7-12 are rejected under 35 U.S.C. §112 in reference to the phrase "the second memory device."

The Applicant notes the phase "the second memory device" is not found in claim 7, and claim 8 has been previously cancelled. Claim 9, however, does recite "the second memory device" and thus the Applicant assumes the rejection is directed thereto. The Applicant has amended claim 9 and it now reads "***a*** second memory device." Support for this claim element may be found in the example "second memory structure" 320, shown in Fig. 3, among other places.

Claim Rejections - 35 U.S.C. § 103(a)

At pages 3-6 of the Office Action, claims 1-7, 9-12, and 21-41 were cited under 35 U.S.C. § 103(a) in regards to Fritchman, U.S. Patent No. 6,785,677 B1, issued on Aug. 31, 2004 (hereinafter "Fritchman") in view of Sherman, U.S. Patent No. 6,389,507 B1, issued on May 14, 2002 (hereinafter "Sherman").

Claims 36-41 are discussed above.

Claim 1, representative in part of claims 1-7, 2-12 and 21-35, recites:

> 1. A method for operating a pattern matching engine having a plurality of information storage entries with two or more regular expressions, each regular expression including a plurality of characters and each regular expression associated with a corresponding action to be applied when matching strings are found, the method comprising the steps of:
>
> identifying one or more borders within each regular expression, the one or more borders separating the regular expression into a plurality of sub-expressions, each sub-expression having a plurality of sequential characters;
>
> loading each of a plurality of entries of the pattern matching engine with the plurality of the sequential characters from one of the sub-expressions, wherein the borders are defined by a predetermined sequence

of regular expression metacharacters, and the entries stored in content addressable memory (CAM);

applying a string from a network message to the entries of the pattern matching engine to search the string simultaneously for sub-expressions from each of the two or more regular expressions;

determining that the sub-expressions of at least one regular expression match the string; and

executing the corresponding action associated with that at least one regular expression on the network message.

Fritchman discloses a client submitting SQL queries to a server, which then searches a database for any records matching the SQL queries. *See* Fritchman, Fig. 1; col. 9, lines 14-26. The SQL queries are apparently searched for one at a time. *See* Fig. 2 (noting only 1 expression, namely "ABC%DEF%XYZ" searched for). For each SQL query, Fritchman divides the SQL query into a prefix, a suffix, and an interior segment for separate processing of each. *See* col. 3, lines 49-53. "[T]he pattern [of the prefix, suffix, or interior segment] is applied to each database value in turn." *See* Fritchman, col. 8, lines 13-14 (emphasis added).

Sherman discloses a method for using a RAM as a CAM, to reduce the cost and increase the packaging efficiency of the CAM. *See* Sherman, col. 2, lines 31-40.

Neither reference suggests the Applicant's claimed "***applying a string from a network message to the entries of the pattern matching engine to search the string simultaneously for sub-expressions from each of the two or more regular expressions.***"

The Applicant claims a novel technique that ***simultaneously*** searches a string for ***sub-expressions from each of the two or more regular expressions.*** Rather than simultaneously search for sub-expressions from multiple regular expressions, Fritchman looks at to sub expressions of **one SQL query**. Indeed, Fritchman even suggests matching sub expressions of the one **SQL query** themselves one at a time, i.e. sequentially. Thus, the sequential searching in Fritchman is far different than what the Applicant claims.

Furthermore, the deficiencies in Fritchman are not remedied by combination with Sherman. Sherman simply discloses a CAM. While a CAM generally allows one to

match a string against several entries within the CAM, simply having a CAM does not suggest what the Applicant's claims. The Applicant specifically claims searching ***a string simultaneously for sub-expressions from each of the two or more regular expressions.*** There is no discussion in Sherman that a CAM should be loaded with sub-expressions ***from each of the two or more regular expressions.***

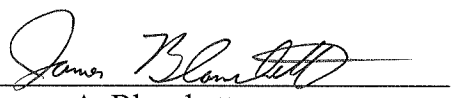
Therefore, Applicant respectfully urges that Fritchman and Sherman, either taken singly or taken in combination, are legally insufficient to render the claims obvious under 35 U.S.C. § 103(a) because of the absence from both references of "***applying a string from a network message to the entries of the pattern matching engine to search the string simultaneously for sub-expressions from each of the two or more regular expressions.***"

Should the Examiner believe a telephonic interview would be helpful in the disposition of this Application, the Examiner is encouraged to call the undersigned attorney at (617) 951-2500.

In summary, all the independent claims are believed to be in condition for allowance and therefore all dependent claims that depend there from are believed to be in condition for allowance. The Applicant respectfully solicits favorable action.

Please charge any additional fee occasioned by this paper to our Deposit Account No. 03-1237.

Respectfully submitted,



James A. Blanchette
Reg. No. 51,477
CESARI AND MCKENNA, LLP
88 Black Falcon Avenue
Boston, MA 02210-2414
(617) 951-2500